SYSTEM AND METHOD FOR ENCODING PACKET HEADER TO ENABLE HIGHER BANDWIDTH EFFICIENCY ACROSS PCIe LINKS

Inventor: William P. Tsu, San Jose, CA (US)

Correspondence Address:
PATTERSON & SHERIDAN, L.L.P.
3040 POST OAK BOULEVARD
SUITE 1500
HOUSTON, TX 77056 (US)

Assignee: NVIDIA Corporation

Appl. No.: 11/253,870
Filed: Oct. 19, 2005

Publication Classification
(51) Int. Cl.
G06F 13/38 (2006.01)
(52) U.S. Cl. .................................................. 710/62

ABSTRACT
A computer system that employs Peripheral Component Interconnect Express (PCIe) links includes devices that generate a PCIe packet having a header portion that is smaller than the header portion for a conventional PCI packet. The devices may be an endpoint device, such as a graphics processor, and a chipset, such as a root-complex. The reduced size header improves the bus throughput efficiency of the computer system and reduces power requirements for the computer system.
Start

Receive a memory read request

Read data from a memory location in accordance with the memory read request

Generate a read completion packet

Transmit the read completion packet

Stop

FIG. 2
Start

Store at least one requisite parameter associated with a memory read request in a tracking table 302

Index the tracking table with a memory-read tag 304

Transmit a PCIe packet to the second device 306

Stop

FIG. 3
### FIG. 4A

<table>
<thead>
<tr>
<th>7 6 5 4 3 2 1 0</th>
<th>7 6 5 4 3 2 1 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>+0</td>
<td>+1</td>
</tr>
<tr>
<td>byte0&gt; R TLP seq number</td>
<td>Data Link Layer Header</td>
</tr>
<tr>
<td>byte2&gt; R Fmt Type R TC R</td>
<td>Transport Layer Header</td>
</tr>
<tr>
<td>byte4&gt; T E R O N S R</td>
<td></td>
</tr>
<tr>
<td>byte5&gt; Compl Complete ID</td>
<td></td>
</tr>
<tr>
<td>byte8&gt; Status B C</td>
<td>Byte Count</td>
</tr>
<tr>
<td>byte10&gt; Requester ID</td>
<td></td>
</tr>
<tr>
<td>byte12&gt; Tag R Lower Address</td>
<td></td>
</tr>
</tbody>
</table>

### FIG. 4B

<table>
<thead>
<tr>
<th>7 6 5 4 3 2 1 0</th>
<th>7 6 5 4 3 2 1 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>+0</td>
<td>+1</td>
</tr>
<tr>
<td>byte0&gt; Tag[3:0] TLP seq number</td>
<td>Data Link Layer Header</td>
</tr>
</tbody>
</table>
Start

Generate a memory write request packet

Transmit the memory write request packet

Stop

FIG. 5
### FIG. 6A

<table>
<thead>
<tr>
<th>+0</th>
<th>+1</th>
</tr>
</thead>
<tbody>
<tr>
<td>7 6 5 4 3 2 1 0</td>
<td>7 6 5 4 3 2 1 0</td>
</tr>
<tr>
<td>byte0&gt; R</td>
<td>TLP seq number</td>
</tr>
<tr>
<td>byte2&gt; R Fmt</td>
<td>Type R TC R</td>
</tr>
<tr>
<td>byte4&gt; T D P O S R</td>
<td>Length R</td>
</tr>
<tr>
<td>byte6&gt; Req ID</td>
<td></td>
</tr>
<tr>
<td>byte8&gt; Tag</td>
<td>Last DW BE 1st DW BE</td>
</tr>
<tr>
<td>byte10&gt; Address[31:16]</td>
<td></td>
</tr>
<tr>
<td>byte12&gt; Address[15:2]</td>
<td>R</td>
</tr>
</tbody>
</table>

**Data Link Layer Header**

**Transport Layer Header**

### FIG. 6B

<table>
<thead>
<tr>
<th>+0</th>
<th>+1</th>
</tr>
</thead>
<tbody>
<tr>
<td>7 6 5 4 3 2 1 0</td>
<td>7 6 5 4 3 2 1 0</td>
</tr>
<tr>
<td>byte0&gt; 1 1 TC[1:0]</td>
<td>TLP seq number</td>
</tr>
<tr>
<td>byte2&gt; Address[31:16]</td>
<td></td>
</tr>
<tr>
<td>byte4&gt; Address[15:2] R O S</td>
<td></td>
</tr>
</tbody>
</table>

**Data Link Layer Header**

**Transport Layer Header**
FIG. 7

Endpoint device

Tracking-table module

Chipset
SYSTEM AND METHOD FOR ENCODING PACKET HEADER TO ENABLE HIGHER BANDWIDTH EFFICIENCY ACROSS PCIe LINKS

BACKGROUND OF THE INVENTION

0001. Field of the Invention

The invention generally relates to communication in computer systems. More specifically, the invention relates to a method and system for communication in a computer system that employs Peripheral Component Interconnect Express (PCIe) links.

0003. Description of the Related Art

0004. The Peripheral Component Interconnect Express (PCIe) is a general purpose Input/Output (I/O) interconnect used for communication between two or more devices inside a computer. The examples of the devices may include a graphics processor and a chipset. A computer system that employs PCIe communicates by sending packets. The packets are formed in three discrete logical layers that include the transaction layer, the data link layer, and the physical layer. Each packet has a header corresponding to these layers and a data payload portion. The header contains information that may include format, type, and attributes of a packet, address/routing information, encoding information, and data protection. The data payload portion contains data required by a device.

0005. In conventional systems, a read completion packet is generated by a chipset that comprises a root-complex, upon completion of a memory read request issued by an endpoint device. The read completion packet includes a conventional read completion header portion and a data payload portion. The conventional read completion header portion includes a physical layer header that is two bytes long, a data link layer header that is six bytes long and a transaction layer header that is twelve bytes long.

0006. Additionally, in conventional systems, a memory write request packet is generated by the endpoint device when issuing a memory write request to the root-complex. The memory write request packet includes a conventional memory write request header portion and a data payload portion. The conventional memory write request header portion includes a physical layer header that is two bytes long, a data link layer header that is six bytes long and a transaction layer header that is twelve bytes long.

0007. These packets consume a large amount of bus bandwidth of computer systems employing PCIe links and limit the bus throughput efficiency of such systems. There is therefore, a need for a method and system that can increase the bus throughput efficiency of computer systems employing PCIe links.

SUMMARY OF THE INVENTION

0008. An aspect of the invention is to provide a method and system for improving the bus throughput efficiency of computer systems employing PCIe links.

0009. In order to achieve the above aspect, methods and systems for communication in a computer system that employs Peripheral Component Interconnect Express (PCIe) links are provided.

0010. In an embodiment of the invention, a method of processing a memory read request is provided. The method includes receiving the memory read request over the PCIe link and reading data from a memory location in accordance with the memory read request. The method further includes generating a read completion packet including a header portion and a data payload portion, wherein the header portion is less than 20 bytes long. The read completion packet is then transmitted over the PCIe link.

0011. In another embodiment of the invention, a method of issuing a memory write request over the PCIe link is provided. The method includes generating a memory write request packet including a header portion and a data payload portion. The header portion is less than 20 bytes long. The memory write request is then transmitted over the PCIe link.

BRIEF DESCRIPTION OF THE DRAWINGS

0012. The accompanying figures, where like reference numerals refer to identical or functionally similar elements throughout the separate views and which together with the detailed description below are incorporated in and form part of the specification, serve to further illustrate various embodiments and to explain various principles and advantages all in accordance with the invention:

0013. FIG. 1 is a block diagram showing an environment (that is exemplary), in which various embodiments of the invention can function.

0014. FIG. 2 is a flowchart of method steps for processing a memory read request in a computer system.

0015. FIG. 3 is a flowchart of method steps for issuing the memory read request, in accordance with an embodiment of the invention.

0016. FIG. 4A shows a header portion of a prior art read completion packet.

0017. FIG. 4B shows a header portion of a read completion packet that is generated in accordance with an exemplary embodiment of the invention.

0018. FIG. 5 is a flowchart of method steps for issuing a memory write request in a computer system.

0019. FIG. 6A shows a header portion of a prior art memory write request packet.

0020. FIG. 6B shows a header portion of a memory write request packet that is generated in accordance with an exemplary embodiment of the invention.

0021. FIG. 7 is a block diagram showing a computer system, in accordance with an embodiment of the invention.

DETAILED DESCRIPTION

0022. Various embodiments of the invention provide methods and systems for communication in a computer system that employs Peripheral Component Interconnect Express (PCIe) links. The PCIe is a general purpose Input/Output (I/O) interconnect used for communication between two or more devices inside a computer. The devices may include one or more endpoint devices and a chipset.

0023. FIG. 1 is a block diagram showing an environment (that is exemplary), e.g., a computer system, in which various embodiments of the invention can function. Envi-
Environment 100 includes a chipset 102, an endpoint device 104 and an endpoint device 106. It will be apparent to a person skilled in the art that the number of endpoint devices does not limit the invention in any way.

[0024] Chipset 102 is a root-complex, e.g., a chipset comprising chips that are commonly known as northbridge and southbridge that communicates with at least one of endpoint device 104 and endpoint device 106, a Central Processing Unit (CPU) 108 and a memory 110 for storing information that may be required by the different units. Chipset 102 is connected to endpoint device 104 and endpoint device 106 through PCIe links. Each of chipset 102, endpoint device 104 and endpoint device 106 is provided with a specially configured PCIe interface through which the PCIe link connection is made. Chipset 102 may also communicate with endpoint device 104 and endpoint device 106 through a switch that is connected to each of chipset 102, endpoint device 104 and endpoint device 106 by PCIe links.

[0025] Endpoint device 104 and endpoint device 106 may comprise I/O devices. Examples of these endpoint devices may include, but are not limited to, a graphics processor, an ethernet card, and a sound card. The communication is carried out by sending a plurality of PCIe packets on PCIe links that connect the endpoint devices 104, 106 to chipset 102.

[0026] A PCIe packet is generally formed in discrete logical layers such as a transaction layer, a data link layer and a physical layer. Each PCIe packet contains a header corresponding to one or more of these discrete logical layers, i.e., a Transaction Layer Header (TLH), a Data Link Layer Header (DLLH), and a Physical Layer Header (PLH). The data contained in these headers include, but are not limited to, format, type, attribute of a PCIe packet, address/routing information, encoding information, data protection and length of data payload. In addition to these headers, each PCIe packet includes a data payload. The data payload contains the data that is to be used by a logical device that receives the PCIe packet. Methods for generating a PCIe packet for communication in a PCIe system in accordance with various embodiments of the invention are explained hereinafter.

[0027] FIG. 2 is a flowchart of method steps for processing a memory read request in a computer system, in accordance with an embodiment of the invention. The computer system comprises a first device and a second device that are coupled to each other by a PCIe link. The first device may be one of endpoint devices 104, 106 and the second device may be chipset 102. The first device issues the memory read request. In an embodiment of the invention, the first device stores at least one requisite parameter in a tracking table indexed by a memory-read tag. The method of storing at least one requisite parameter is explained in conjunction with FIG. 3.

[0028] The memory read request is received by the second device at step 202. At step 204, data is read by the second device from a memory location in accordance with the memory read request. Examples of the memory location may include, but are not limited to, Random Access Memory (RAM), Read Only Memory (ROM), and Dynamic Random Access Memory (DRAM). At step 206, a read completion packet is generated upon completion of the memory read request. Thereafter, at step 208, the read completion packet is transmitted over the PCIe link to the first device.

[0029] The read completion packet includes a data payload portion and a header portion. The data payload portion includes information that is requested by the first device in the memory read request. The header portion is less than 20 bytes long. In an exemplary embodiment of the invention, the header portion includes a PH that is two bytes long and a DLLH that is six bytes long. As a result, in this exemplary embodiment of the invention, the header portion of the read completion packet is eight bytes long.

[0030] In an embodiment of the invention, on the header portion of the read completion packet, a Transaction Layer Packet (TLP) sequence number and a completion tag are transmitted. The TLP sequence number is the identification number for a PCIe packet. The completion tag references a tracking table that contains the at least one requisite parameter associated with the memory read request.

[0031] FIG. 3 is a flowchart of method steps for issuing the memory read request, in accordance with an embodiment of the invention. At step 302, the first device stores at least one requisite parameter associated with the memory read request in a tracking table. Examples of the at least one requisite parameter may include, a Traffic Class (TC) field, a Relaxed Ordering (RO) field, a No Snoop (NS) field, a byte count field, and a lower address field. The TC field classifies a PCI transaction sequence into different classes. Each class associates a PCI transaction sequence with the type of service required by it. The RO field and the NO field are attributes that help in handling traffic on a PCIe link. The values in these fields must remain the same as a memory read request is carried out and a read completion is reported. The byte count field conveys the numbers of data bytes that are present in a packet. The lower address field is the lowest seven bits of the read address. The complete read address is specified in the header portion of a memory read request packet. At step 304, the tracking table is indexed with a memory-read tag that is transmitted with the memory read request issued by the first device. The memory-read tag is an eight-bit number. In the embodiment of the present invention illustrated herein, the memory-read tag has a value that is less than 16. At step 306, a PCIe packet is then transmitted to the second device by the first device to issue the memory read request.

[0032] FIG. 4A shows a header portion 402 of a prior art read completion packet. Header portion 402 includes a PH (not shown in FIG. 4A) that is two bytes long, a DLLH that is six bytes long, including the two bytes shown in FIG. 4A and four bytes of Link Cyclic Redundancy Check (CRC) code not shown in FIG. 4A, and a TLH that is twelve bytes long. The DLLH of header portion 402 includes a zeroth byte and a first byte. The TLH of header portion 402 includes bytes from a second byte to a 13th byte. A TLP sequence number is transmitted on the first four bits of the zeroth byte of header portion 402 concatenated with eight bits of the first byte. The last four bits of the zeroth byte are transmitted as reserved bits. Further, a first set of parameters is transmitted on the TLH of header portion 402. The first set of parameters include a format field, a type field, a memory-tag field, a TC field, a Transaction Description (TD) field, an Endpoint (EP) field, an RO field, an NS field, a complete ID, a requestor ID, a completion status, a lower address field, a length field, a BC field, and a byte count field.

[0033] The format field describes the format of a request. The type field describes the type of a request. Further, the
memory-tag field of header portion 402 is eight bits long. The TD field describes the properties of a PCIe transaction sequence. The EP field is used for error forwarding. The length field contains a value corresponding to the total number of bytes that have been read in response to the read request. The completer ID is a combination of a completer’s bus number, device number, and function number used to uniquely identify the completer. The requester ID is a combination of a requester’s bus number, device number, and function number that uniquely identifies the requester. The completion status indicates whether or not the requested read was successful. The BC field is typically zero.

[0034] In addition to the first set of parameters, a plurality of bits of the TLH of header portion 402 is transmitted as reserved bits. For example, the seventh bit of the second byte, the zeroth bit to the third bit of the third byte and the seventh bit of the third byte, the second bit and the third bit of the fourth byte, and the seventh bit of the 13th byte of the TLH of header portion 402 are transmitted as reserved bits.

[0035] FIG. 4B shows a header portion 404 of a read completion packet that is generated in accordance with an exemplary embodiment of the invention. Header portion 404 includes a PLH (not shown in FIG. 4B) that is two bytes long and a DILH that is six bytes long, including the two bytes that are shown in FIG. 4B and four bytes of LRC code not shown in FIG. 4B. The TLP sequence number is transmitted on the eight bits of the first byte concatenated with the first four bits of the zeroth byte of the DILH of header portion 404. The completion tag is four bits long and is transmitted on the last four bits of the zeroth byte of the DILH of header portion 404. The completion tag includes the four least significant bits of the memory-read tag that is transmitted with the memory read request. In the embodiment of the present invention illustrated herein, the four most significant bits of the memory-read tag are assumed to be zero and are not included in the header portion 404.

[0036] The first device receives the read completion packet with header portion 404 and extracts the header information in the following manner. First, the format and type of read completion packet are not included in header portion 404 but read completion packets are of only one type, namely read completion, and so the PCIe interface of the first device assumes the received PCIe packet to be of the read completion type. There are other header information that are assumed to have certain values and therefore are not included in header portion 404. They include: the TD field, which is assumed to be zero; the EP field, which is assumed to be zero; the completion status, which is assumed to be successful; the BC field, which is assumed to be zero; and the four most significant bits of the memory-tag field, which are assumed to be zero. Further, the completer ID and the requester ID are not used and so these fields are not included in header portion 404. The length field is not included in header portion 404 because the packet length is calculated from the received packet. The remaining header information, including the TC field, RO field, NS field, the byte count field, and the lower address field are recovered from the tracking table using the completion tag.

[0037] FIG. 5 is a flowchart of method steps for issuing a memory write request in a computer system, in accordance with an embodiment of the invention. The computer system includes a first device and a second device that are coupled to each other by a PCIe link. The first device may be one of endpoint devices 104, 106 and the second device may be chipset 102. The memory write request is issued by the first device. At step 502, a memory write request packet is generated. The memory write request packet includes a data payload portion and a header portion that is less than 20 bytes. The data payload portion contains information that is required by the second device for processing the memory write request. In an exemplary embodiment of the invention, the header portion includes a PLH that is two bytes long, a DILH that is six bytes long, and a TLH that is four bytes long. As a result, in this exemplary embodiment, the header portion is 12 bytes long.

[0038] Further, at step 504, the memory write request packet is transmitted over the PCIe link to the second device. A TLP sequence number and a predetermined set of parameters are transmitted on the header portion of the memory write request packet.

[0039] The predetermined set of parameters includes one or more of a format indicator, a size indicator, a TC field, a RO field, and an NS field. Further, the predetermined set of parameters includes a memory write address.

[0040] FIG. 6A shows a header portion 602 of a prior art memory write request packet. Header portion 602 includes a PLH (not shown in FIG. 6A) that is two bytes long, a DILH that is six bytes long, including two bytes that are shown in FIG. 6A and four bytes of LRC code not shown in FIG. 6A, and a TLH that is twelve bytes long. The DILH of header portion 602 includes a zeroth byte and a first byte and the TLH of header portion 602 includes bytes from a second byte to a 13th byte. A TLP sequence number is transmitted on the eight bits of the first byte concatenated with first four bits of the zeroth byte. The last four bits of the zeroth byte are transmitted as reserved bits. Other parameters include a format field, a type field, a memory-write tag field, a 3-bit TC field, a TD field, an EP field, an RO field, an NS field, a length field, a requester ID field, last Double Word Byte Enable (Last DW BE) field and a last DW BE field. The last DW BE field contains byte enables for a last DW of a service request. The last DW BE field contains byte enables for a first DW of a service request. Further, address bits are transmitted on the tenth byte, twelfth byte, and the last six bits of the thirteenth byte. The seventh bit of second byte, the zeroth bit to third bit of third byte and the seventh bit of the third byte, second and third bit of the fourth byte, and the zeroth bit and the first bit of the 13th byte are transmitted as reserved bits.

[0041] FIG. 6B shows a header portion 604 of a memory write request packet that is generated in accordance with an exemplary embodiment of the invention. Header portion 604 includes a PLH (not shown in FIG. 6B) that is two bytes long, a DILH that is six bytes long, including two bytes that are shown in FIG. 6B and four bytes of LCR code not shown in FIG. 6B, and a TLH that is four bytes long. The DILH includes a zeroth byte and a first byte. The TLH includes a second byte, a third byte, a fourth byte and a fifth byte.

[0042] The TLP sequence number and the predetermined set of parameters are transmitted on header portion 604. The TLP sequence number is transmitted on the first byte con-
cated with four bits of the zeroth byte. The predetermined set of parameters includes the format indicator, the size indicator, first bit and second bit of the TC field, the RO field and the NS field. The format indicator is transmitted on the seventh bit of the zeroth byte of the DLLH and it contains a value of one that corresponds to the memory write request. The size indicator that is transmitted on the sixth bit of the zeroth byte of the DLLH contains a value of one that corresponds to a size of 64 bits for the memory write request packet. Further, the first bit and second bit of the TC field is transmitted on the fifth and fourth bit of the zeroth byte. In addition to these parameters, address bits are transmitted on the second, third, and fourth byte concatenated with five bits of the fifth byte. Further, the RO field is transmitted on the first bit and the NS field is transmitted on the zeroth bit of the fifth byte.

[0043] The second device receives the memory write request and extracts the header information in the following way. The format and type are encoded as 1 and 1 to indicate a 64-bit memory write request and as 1 and 0 to indicate a 32-bit memory write request. There is other header information that is assumed to have certain values and thus is not included in header portion 604. They include: the third bit of the TC field, which is assumed to be zero; the TD field, which is assumed to be zero; the EP field, which is assumed to be zero; and each of the last DW BE field and the 1st DW BE field, which are assumed to have a value of one. Further, the requester ID and the memory-write tag are not used and are not included in header portion 604.

[0044] FIG. 7 is a block diagram showing a computer system 700, in accordance with the above invention. Computer system 700 includes a device that has a PCIe interface programmed to generate a PCIe packet having a header portion that is less than 20 bytes. The device may be one of an endpoint device 702 and a chipset 704.

[0045] In an embodiment of the invention, the PCIe packet may be a memory write request packet generated by endpoint device 702 in connection with a memory read request. A predetermined set of parameters and a TLP sequence number are transmitted on the memory write request packet that is received by chipset 704. The predetermined set of parameters is then extracted by the PCIe interface of chipset 704. In another embodiment of the invention, the PCIe packet may be a read completion packet generated by chipset 704 when a memory read request issued by endpoint device 702 has completed. A completion tag and a TLP sequence number are transmitted on the read completion packet that is received by endpoint device 702. The header information is extracted by the PCIe interface of endpoint device 702.

[0046] Endpoint device 702 may include a tracking-table module 706 that stores at least one requisite parameter associated with a memory read request in a tracking table. The at least one requisite parameter is recovered by endpoint device 702 using the tracking table when it receives the read completion packet from chipset 704 in response to the memory read request.

[0047] Various embodiments of the invention provide a method and system for improving the bus throughput efficiency of a PCIe link. For example, for a completion packet with 32 bytes of payload, the bus throughput efficiency improves from 61% to 80%. Similarly, for the completion packet with 64 bytes of payload, the bus throughput efficiency improves from 76% to 89%. In another example, for a posted write packet with 32 bytes of payload, the bus throughput efficiency improves from 61% to 73%. Similarly, for the posted write packet with 64 bytes of payload, the bus throughput efficiency improves from 76% to 84%.

[0048] Further, the computer system implementing various aspects of the invention switches to a lower power state quicker and stays in the low power state for a long time. This contributes to power savings, which is valuable in mobile applications.

[0049] While the invention has been described in conjunction with specific embodiments thereof, additional advantages and modifications will readily occur to those skilled in the art. The invention, in its broader aspects, is therefore not limited to the specific details, representative apparatus, and illustrative examples shown and described. Various alterations, modifications and variations will be apparent to those skilled in the art in light of the foregoing disclosure. Thus, it should be understood that the invention is not limited by the foregoing description, but embraces all such alterations, modifications and variations in accordance with the spirit and scope of the appended claims.

What is claimed is:

1. A computer system having a first device and a second device coupled to each other by a Peripheral Component Interconnect Express (PCIe) link, a method of processing a memory read request by the second device, the method comprising the steps of:
   a. receiving the memory read request issued by the first device over the PCIe link;
   b. reading data from a memory location in accordance with the memory read request;
   c. generating a read completion packet including a header portion and a data payload portion, wherein the header portion is less than 20 bytes long; and
   d. transmitting the read completion packet over the PCIe link to the first device.
2. The method of claim 1, wherein the first device is an endpoint device and the second device is a chipset.
3. The method of claim 1, wherein the header portion is eight bytes long.
4. The method of claim 1, wherein a Transaction Layer Packet (TLP) sequence number and a completion tag are transmitted on the read completion packet.
5. The method of claim 4, wherein the completion tag is used to recover at least one requisite parameter associated with the memory read request.
6. The method of claim 5, wherein the at least one requisite parameter is stored in a tracking table.
7. The method of claim 6, wherein the tracking table is indexed with a memory-read tag.
8. In a computer system having a first device and a second device coupled to each other by a Peripheral Component Interconnect Express (PCIe) link, a method of issuing a memory write request by the first device, the method comprising the steps of:
a. generating a memory write request packet including a header portion and a data payload portion, wherein the header portion is less than 20 bytes long; and

b. transmitting the memory write request packet over the PCIe link to the second device.

9. The method of claim 8, wherein the first device is an endpoint device and the second device is a chipset.

10. The method of claim 8, wherein the header portion is 12 bytes long.

11. The method of claim 8, wherein a Transaction Layer Packet (TLP) sequence number and a predetermined set of parameters are transmitted on the memory write request packet.

12. The method of claim 11, wherein the predetermined set of parameters includes a memory write address.

13. The method of claim 11, wherein the predetermined set of parameters comprises at least one of:

   a. a format indicator, the format indicator indicating a format and type of packet;

   b. a size indicator, the size indicator indicating the size of the packet;

   c. a Traffic Class (TC) field, the TC field classifying PCI transaction sequences into different classes; and

   d. at least one of a Relaxed Ordering (RO) field and a No Snoop (NS) field.

14. A computer system comprising a device that is programmed to generate a Peripheral Component Interconnect Express (PCIe) packet including a header portion, the header portion being less than 20 bytes long.

15. The computer system of claim 14, wherein the device is an endpoint device.

16. The computer system of claim 15, wherein the PCIe packet is a memory write request packet, and the memory write request packet is generated by the endpoint device in connection with a memory write request, and wherein a Transaction Layer Packet (TLP) sequence number and a predetermined set of parameters are transmitted on the memory write request packet.

17. The computer system of claim 15, wherein the endpoint device comprises a tracking-table module, the tracking-table module storing at least one requisite parameter associated with a memory read request in a tracking table, the at least one requisite parameter being recovered by the endpoint device when it receives a read completion packet in response to the memory read request.

18. The computer system of claim 14, wherein the device is a chipset.

19. The computer system of claim 18, wherein the PCIe packet is a read completion packet and the read completion packet is generated by the chipset in connection with a completion of a memory read request, and wherein a Transaction Layer Packet (TLP) sequence number and a completion tag are transmitted on the read completion packet.

20. The computer system of claim 14, wherein the device is a graphics processor.

* * * * *